A Probabilistic Approach for Complete Coverage Path Planning with low-cost Systems

Nils Rottmann[1], Robin Denz[1], Ralf Bruder[1], Elmar Rueckert[1]

***Abstract*—Domestic robots, such as vacuum cleaners or lawn mowers, are mostly based on a low-cost design to make them affordable for the consumer. This often results in such robots being equipped with only simple sensors, such as in-/outside area detectors for lawn mowers. Intelligent navigation and planning strategies, however, usually require additional sensors like LiDAR sensors, cameras or time of flight sensors. Thus, there is a lack of intelligent approaches for the complete coverage of the workspace under consideration of only minimal sensing capabilities.**

In this work, we propose a probabilistic planning method for low-cost robots with limited sensing capabilities to completely cover an enclosed environment. Our planning approach thereby utilizes Monte Carlo Localization for estimating coverage probabilities based on the particle distribution. These coverage probabilities are stored in a grid map on the basis of which an intelligent path planning approach determines the next locations to be visited.

We demonstrate our approach in different simulation scenarios for a realistic autonomous lawn mower with only in-/outside area detection capabilities. As comparison benchmark we use the common random walk mowing pattern.

I. Introduction

Low-cost robots for domestic use, for example lawn mowers, often only employ simplistic random navigation strategies. This is because there sensory capabilities are limited due to their low-cost design to only detect in-/outside area estimations. For example, most common autonomous lawn mowers today use bounding wire, electro-magnetic field measurement technology which safely detects wire crossing [1]. In [2, 3], we proposed an active chlorophyll fluorescence sensory system for replacing the bounding wire to reduce installation and maintenance costs. However, the autonomous lawn mower is still only capable of in-/outside area detection. Based on those measurements, random walk strategies are currently employed to cover the complete working area which are inefficient with respect to time and energy consumption and often get stuck in narrow corridors. Intuitively, common known coverage patterns such as stripes or cycles would be preferable and highly efficient but can not be performed by autonomous lawn mowers due to their uncertain pose estimate. Thus, a trade-off between a perfect structured coverage pattern and random walk is required which accounts for the position uncertainties, reduces working time and covers the complete working area with high probability.

[1]Institute for Robotics and Cognitive Systems, University of Luebeck, Ratzeburger Allee 160, 23562 Luebeck, Germany {rottmann, bruder, rueckert}@rob.uni-luebeck.de, robin.denz@student.uni-luebeck.de

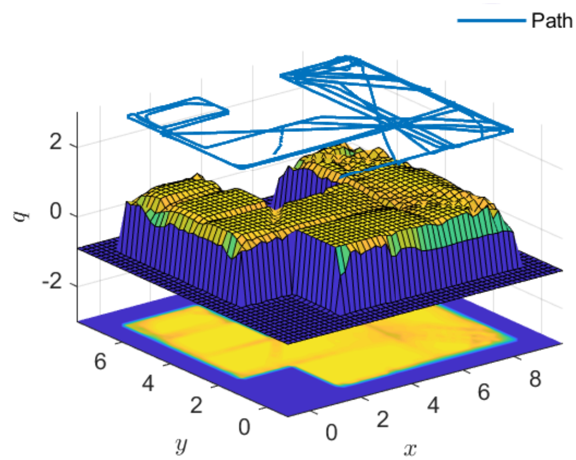


Fig. 1: Example of the proposed path planning method showing the neural activities which force the robot and the path driven by the robot. The coverage rate of the working area is $50\,\%$. The underlying method is described in detail in Section II.

A. Related Work

In the past, many complete coverage path planning (CCPP) strategies have been proposed. Those strategies can be partitioned into cellular decomposition, landmark-based or grid-based methods. For cellular decomposition methods, the free-space of the working area is divided into individual, non-overlapping cells such that the free-space is completely filled. Those cells are then covered by generating an efficient path through all cells, e.g. a ”mowing the lawn“ pattern. Two prominent examples are given with [4] and [5]. In the former, the problem of complete terrain acquisition with arbitrary shaped obstacles has been addressed which produced the ”Seed Spreader“ algorithm. In the latter, the boustrophedon cell decomposition (BCD) has been developed which allows for non-polygonal obstacles within the operation space.

Landmark-based approaches use topological maps detecting natural landmarks for navigation and planning. Those methods are still utilizing the boustrophedon cell decomposition (BCD). Such an algorithm has been introduced in [6], where cellular decomposition is used for coverage path planning by generating a planar graph G with a set of Nodes N and edges E. The overall algorithm is then designed as a finite state machine given the three states ”boundary“, ”normal“ and ”travel“. The method shows coverage accuracies of $99\,\%$ in simulation and $85\,\%$ for real robots.

Fig. 2: The autonomous lawn mower used for learning the model parameters to determine a realistic simulation setting.

Another famous class of CCPP algorithms utilizes grid maps which have been firstly introduced in [7]. Those grid maps are simple to create and to maintain but suffer from exponential growth. One of the first methods applying CCPP on grid maps has been given in [8] where an complete coverage path is planned offline. For a detailed overview of CCPP algorithms, we refer to [9].

Other approaches for CCPP focus on certain areas of application. For example, in [10] the authors proposed a CCPP method for agricultural machines, where trajectories are selected which guarantee complete coverage while minimizing overlapping. The best studied robots for CCPP are vacuum cleaners since they are widely used nowadays and they operate in a simplistic indoor setting. A complete setup for this type of robots, including a simple CCPP method, has been introduced in [11]. More advanced studies covering efficient probabilistic robot cleaning strategies have been introduced in [12] and [13]. In the former, dirt grid maps are introduced which are modeled by Poisson Processes. Based on the modeled dirt distribution, a traveling salesman problem (TSP) is solved for optimally cleaning the working space. In the latter, high-confidence cleaning guarantees under uncertainties are studied. Therefore, a particle filter is used to estimate the dirt distribution assigning a particle to each grid cell of the map. These particles are then updated based on random samples from the robots motion model. The robots path is then updated solving again a TSP.

For autonomous lawn mowers, little research considering efficient path planning methods have been done. In [14], this problem is addressed designing different planning methods with respect to minimal time or energy consumption. However, the proposed method requires an exact pose estimate for the robot utilizing real-time positioning system and self-navigation. Such precise positioning is not available for consumer lawn mowers, as these are only equipped with in-/outside area detection and odometry sensors. Also, most of the previously mentioned methods are not applicable to lawn mowers, since they mostly require remote sensing such as sonar. An approach for CCPP considering only contact sensors has been presented in [15]. However, it requires rectilinear structure of the working environment. Those limitations, either that remote sensing techniques are needed or assumptions of the structure of the environment have to be made, require new probabilistic approaches for CCPP on autonomous lawn mowers to be applicable.

B. Contributions and Organization

We propose a probabilistic CCPP approach for closed environments applicable to autonomous lawn mowers and other low-cost systems which can cope with high uncertainties regarding the pose estimates. For this purpose, the neural network based CCPP approach from [16, 17] is adapted and combined with ideas for high-confidence cleaning guarantees for vacuum cleaners from [13]. To the best of our knowledge, this is the first work which addresses directly the CCPP problem for robots with only in-/outside area detection without any further restrictions to the structure of the environment. The contributions of this paper are thereby three-fold: (1) Adaptation of a neural net planning method for highly uncertain pose estimates, (2) a definition of a probabilistic coverage map and (3) a demonstration of the proposed method in different realistic simulation scenarios.

The paper is organized as follows: We start by introducing our method in Section II, beginning with the definition of the underlying dynamical system and the probabilistic coverage map. We proceed with a short repetition to particle filter which then leads to the actual planning algorithm. In Section III, we demonstrate our approach in different challenging simulation scenarios and in Section IV we conclude.

II. Methods

We assume a map of the closed environment is given as a binary occupancy grid $\mathcal{M}$. Such a map can be generated utilizing only the given in-/outside area detectors by following the boundary line as demonstrated in [18]. Given the map of the environment $\mathcal{M}$, a probability occupancy grid map $\mathcal{C}$ is used to model the coverage of the closed environment. Therefore, we use a similar dynamical system model as proposed in [12, 13] for high-confidence cleaning guarantees. To account for the high uncertainty of the pose estimate given only the odometry and the in-/outside area measurements, a particle filter is used to estimate the pose and the coverage respectively. The CCPP approach then uses a neural network where each cell of the coverage map represents a neuron, as introduced in [16, 17]. Based on the neural activity of the neurons and the current pose, the planner chooses the neighboring cell to move to. This, in combination with the probabilistic coverage information, leads to efficient CCPP under high uncertainties.

A. Probabilistic Coverage Map and Dynamical System

Given the map of the environment $\mathcal{M}$, a probability occupancy grid map $\mathcal{C}$ is generated with $c_i \in [0, 1]$ being the probability of the cell i to be covered by the robot initialized as $c_i = 0$, $\forall i$. Let $\boldsymbol{p}_t$ be the pose of robot at time step t, $\boldsymbol{u}_t$ the input signals to the robot, $\boldsymbol{z}_t$ the sensor measurements and $\boldsymbol{c}_t$ the coverage states of all cells, then the path of the robot and the coverage can be defined as joint posterior distribution

$$\begin{aligned}\text{prob}(\boldsymbol{p}_{0:t}, \boldsymbol{c}_{0:t}|\boldsymbol{u}_{1:t}, \boldsymbol{z}_{1:t}) = \eta\, &\underbrace{\text{prob}(\boldsymbol{z}_t|\boldsymbol{p}_t)}_{\text{sensor model}} \\ &\underbrace{\text{prob}(\boldsymbol{c}_t|\boldsymbol{c}_{t-1}, \boldsymbol{p}_{t-1}, \boldsymbol{p}_t)}_{\text{coverage model}}\ \underbrace{\text{prob}(\boldsymbol{p}_t|\boldsymbol{p}_{t-1}, \boldsymbol{u}_t)}_{\text{motion model}} \\ &\underbrace{\text{prob}(\boldsymbol{p}_{0:t-1}, \boldsymbol{c}_{0:t-1}|\boldsymbol{u}_{1:t-1}, \boldsymbol{z}_{1:t-1})}_{\text{prior distribution}}.\end{aligned} \tag{1}$$

In comparison to [13], we do not have a coverage sensor model which gives us additional information for pose estimation. By considering the Markov property $\text{prob}(a_t|a_{t-1:0}) = \text{prob}(a_t|a_{t-1})$, Equation (1) turns into

$$\begin{aligned}\text{prob}(\boldsymbol{p}_t, \boldsymbol{c}_t|\boldsymbol{u}_t, \boldsymbol{z}_t) = \eta\, &\text{prob}(\boldsymbol{z}_t|\boldsymbol{p}_t) \\ &\text{prob}(\boldsymbol{c}_t|\boldsymbol{c}_{t-1}, \boldsymbol{p}_{t-1}, \boldsymbol{p}_t)\ \text{prob}(\boldsymbol{p}_t|\boldsymbol{p}_{t-1}, \boldsymbol{u}_t) \\ &\text{prob}(\boldsymbol{p}_{t-1}, \boldsymbol{c}_{t-1}|\boldsymbol{u}_{t-1}, \boldsymbol{z}_{t-1})\end{aligned} \tag{2}$$

which gives us an iterative update rule for the pose estimate and the coverage based on the current inputs and sensor measurements. In the following we shortly define each of the models introduced in Equation (1).

1) Sensor Model: Our robot is equipped with two in-/outside area detectors, more precisely chlorophyll fluorescence sensors, which give the information whether the sensors are over grass (inside) or not (outside). The sensors are placed at the left and right front of the robot. Based on the current pose $\boldsymbol{p}$, an estimate of the measurements can be made given the information from the map $\mathcal{M}$.

2) Coverage Model: The coverage model reflects the change in coverage based on the movements of the robot. A general probabilistic model can be defined as

$$\begin{aligned}&\text{prob}(\boldsymbol{c}_t|\boldsymbol{c}_{t-1}, \boldsymbol{p}_{t-1}, \boldsymbol{p}_t) = \\ &\quad \text{prob}(\boldsymbol{c}_{t-1}) + P(\boldsymbol{p}_{t-1}, \boldsymbol{p}_t) \cdot (1 - \text{prob}(\boldsymbol{c}_{t-1}))\,,\end{aligned} \tag{3}$$

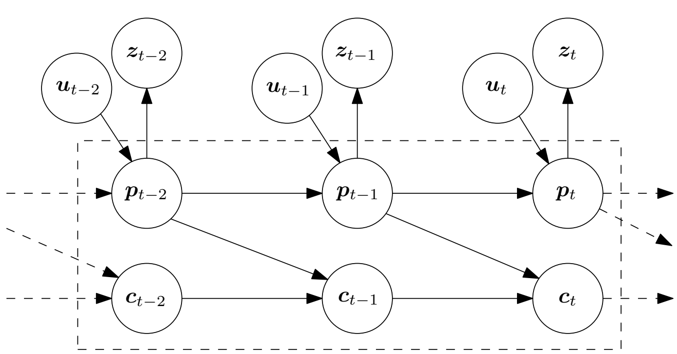
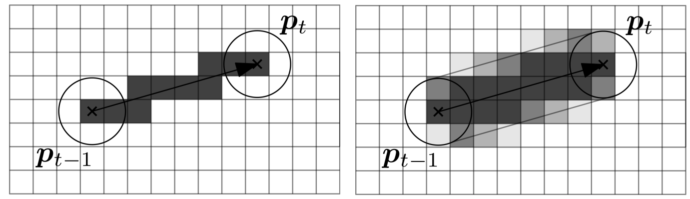


Fig. 3: The dynamical system, Equation (1), illustrated as graph.



(a) Line drawing approach. (b) Unweighted area sampling.

Fig. 4: Different coverage models where the circles represent the robot's effector, e.g. the cutter. The first approach is conservative and computationally fast but underestimates the true coverage whereas the latter is more accurate but comes with higher computational costs and might slightly overestimate the coverage.

where $P(\boldsymbol{p}_{t-1}, \boldsymbol{p}_t)$ is the probability of the robot covering a certain grid cell by moving from pose $\boldsymbol{p}_{t-1}$ to $\boldsymbol{p}_t$. In comparison to [13], where unweighted area sampling is used to determine $P(\boldsymbol{p}_{t-1}, \boldsymbol{p}_t)$ (Figure 4b), we use a more conservative yet computationally faster approach in order to lower the computational burden onto our system (Figure 4a). Our approach is based on simple line drawing algorithms where a cell is marked as covered if the robot passed it. Therefore, the resolution of the coverage grid has to be chosen as

$$\text{Resolution} \geq \frac{2\sqrt{2}}{d}, \tag{4}$$

where d is the diameter of the robot's effector. This ensures the coverage of the whole cell. The coverage probability is then $P(\boldsymbol{p}_{t-1}, \boldsymbol{p}_t) = 1$ if the robot traversed the certain grid cell and $P(\boldsymbol{p}_{t-1}, \boldsymbol{p}_t) = 0$ otherwise.

3) Motion Model: For the motion model we use the in [19] proposed odometry model. Therefore, we determined the model noise parameters $a_1, \ldots, a_4$ applying Maximum Likelihood Estimation [20] on with the real robot recorded data. We validated our results with the Kolmogorow-Smirnow-Test [21]. The parameter values are given in Table II.

B. Particle Filter

To efficiently generate a pose estimate for our robot by fusing the odometry and sensor data, we use a standard particle filter algorithm [19] to handle the binary in-/outside area measurements. The general idea of the particle filter is to represent the probability distribution of the posterior by a set of samples, called particles, instead of using a parametric form as the Kalman Filter does. Here each particle

$$\mathcal{X}_t = \boldsymbol{x}_t^{[1]}, \boldsymbol{x}_t^{[2]}, \ldots, \boldsymbol{x}_t^{[N]} \tag{5}$$

represents a concrete instantiation of the state at time t, where N denotes the number of particles used. The belief $bel(\boldsymbol{x}_t)$ is then approximated by the set of particles $\mathcal{X}_t$. The pose estimate of the robot can be calculated by taking the mean over all

N particles. The Bayes filter posterior is used to include the likelihood of a state hypothesis $\boldsymbol{x}_t$

$$\boldsymbol{x}_t^{[i]} \sim p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}). \tag{6}$$

Here $\boldsymbol{z}_{1:t}$ and $\boldsymbol{u}_{1:t}$ represent the measurement history and the input signal history respectively.

Given the positions of the N particles, the coverage map is updated when the robot traversed from one cell to another. According to the coverage model proposed in Equation (3) and Figure 4a the update rule can be defined as

$$c_{t,i} = c_{t-1,i} + \frac{n_{t,i}}{N}(1 - c_{t-1,i}), \tag{7}$$

where $c_{t,i}$ is the coverage probability of cell i at time t and $n_{t,i}$ the number of particles in cell i at time t.

C. Complete Coverage Path Planner

Given the estimation of the robots trajectory and estimation of the coverage, an efficient path planning scheme is left to define. Due to the high uncertainty of the robot's pose and the associated uncertainties in map coverage, such a path planning scheme requires rapid adaptability. Hence, solving a traveling salesman problem (TSP) on the fully connected graph over all cells of the map as in [13] is not feasible. Instead, we adapt the neural network approach introduced in [16, 17] to uncertain pose and coverage estimates. The neural network approach is thereby derived from the shunting equation [22] which was inspired by a model for a patch of a membrane introduced in [23]. In this approach, neurons are generated which each have a neural activity. These neural activities then provide an attraction on the basis of which the robot plans its next steps.

Let each cell of the coverage map represent a neuron and q_i the neural activity of the neuron i, then the change of neuronal activity can be described according to [16] as

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = -Aq_i + (B - q_i)\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[q_j]^+\right) - (D + q_i)[I_i]^-. \tag{8}$$

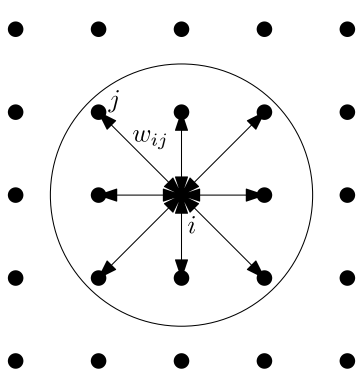


Fig. 5: A schematic of the neural network with the neighborhood of a cell and the cell connections.

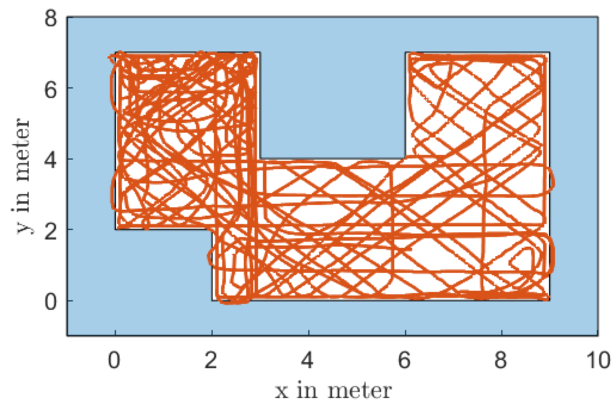


Fig. 6: Example path after reaching 90 % coverage.

Here, A, B and D are non-negative parameters representing the passive decay rate, the upper and the lower bound of the neural activity. The function operators $[a]^+$ and $[a]^-$ are defined as $[a]^+ = \max\{a, 0\}$ and $[a]^- = \min\{-a, 0\}$. The neuronal activities are initialized as $q_i = 0, \forall i$ but receive an external input I_i. Following to [16] and considering the coverage probabilities c_i, we propose

$$I_i = \begin{cases} (1 - c_i)\, E & \text{if cell } i \text{ is inside the working area} \\ -E & \text{if cell } i \text{ is outside the working area} \end{cases}. \tag{9}$$

Here, the external input accelerator E should be chosen such that $E \gg B$. The neurons are each connected to their neighboring cells, as illustrated in Figure 5. The connections weights can be defined as

$$w_{ij} = f(||\boldsymbol{x}_i - \boldsymbol{x}_j||), \tag{10}$$

where $\boldsymbol{x}_i$ represents the location of the i-th neuron. For example, [16] proposes the weighting function

$$f(a) = \begin{cases} \frac{\mu}{a} & \text{if } 0 < a < r \\ 0 & \text{if } a \geq r \end{cases} \tag{11}$$

with μ and r being positive constants. Here, a regular grid is assumed such that we propose a constant symmetric weighting matrix

$$\boldsymbol{W} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} \\ 1 & 0 & 1 \\ \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{12}$$

for efficiently using image filtering techniques for determining the input of the neighboring cells. Since Equation (8) only allows for positive neural activities to propagate between neighboring neurons, negative neural activities stay local. In other words, uncovered areas inside the working area attract the robot globally while areas outside the working space only locally push the robot away. Samples of neural activity distributions over the cells are shown in Figure 7 at two different coverage states.

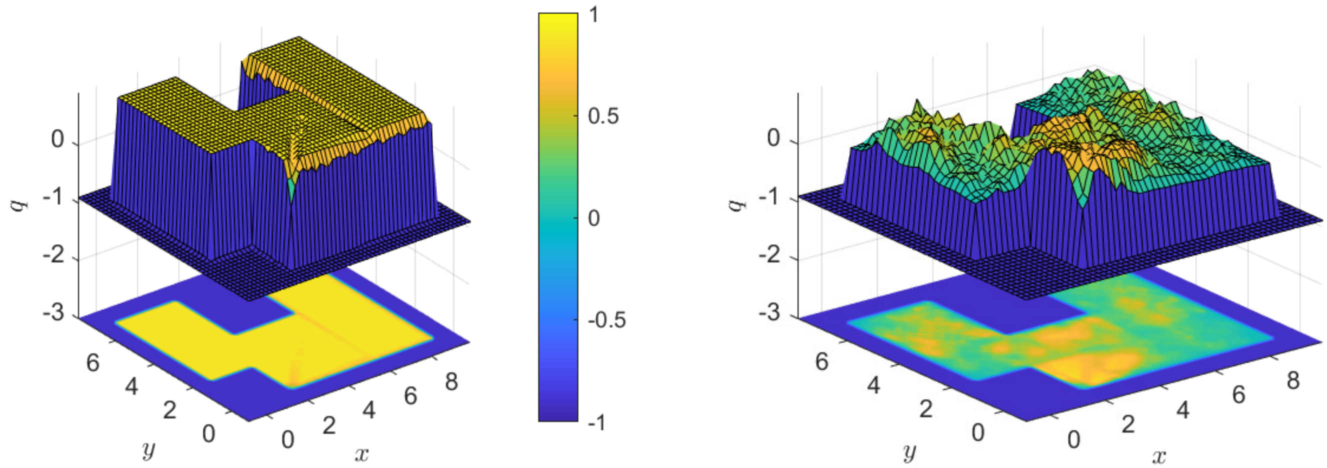


Fig. 7: Neural activities q_i for different coverage values. The left panel shows the neural activities after $10\,\%$ coverage is reached and the right panel the neural activities after $95\,\%$ coverage is reached.

Following [16], the robot's path is planned based on the neural activity landscape. More precisely, let $\boldsymbol{x}_i$ be the robot's current position and q_i the corresponding neural activity, then the next position $\boldsymbol{x}_{\text{next}}$ the robot is sent to can be defined as the position of the neighboring cell with the largest neural activity. Thus,

$$q_{\boldsymbol{x}_{\text{next}}} = \max\left(q_j, j = 1, \ldots, k\right) \tag{13}$$

with k being the number of neighboring cells of q_i. Considering the robot's motion model, we like to avoid unnecessary turns with the robot since turning movements are much more prone to odometry errors. Thus, an additional term to Equation (13) is added which takes the current orientation φ_i of the robot into account

$$q_{\boldsymbol{x}_{\text{next}}} = \max\left(q_j - \gamma g_j, j = 1, \ldots, k\right) \tag{14}$$

with

$$g_j = \left|\varphi_i - \text{atan2}\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)\right| \tag{15}$$

and γ being a parameter which has to be set appropriately. This ensures a preferred next cell allocation based on the robots current pose. In Figure 6, an example path after reaching $90\,\%$ coverage is shown.

D. Relocalization

Since the robot has a large odometry error but only in-/outside area detection sensors, the robot receives only few valuable measurement data during path execution. Thus, it is required that the robot stops path execution and starts to relocalize itself when the pose estimate becomes quite uncertain, thus exceeds a certain variance value σ_{max}. For the relocalization, the robot drives along the boundary to get different measurement signals for improving the accuracy of the pose estimate by the particle filter until the pose estimates variance is below another threshold σ_{min} after which the robot continues with path execution.

III. Results

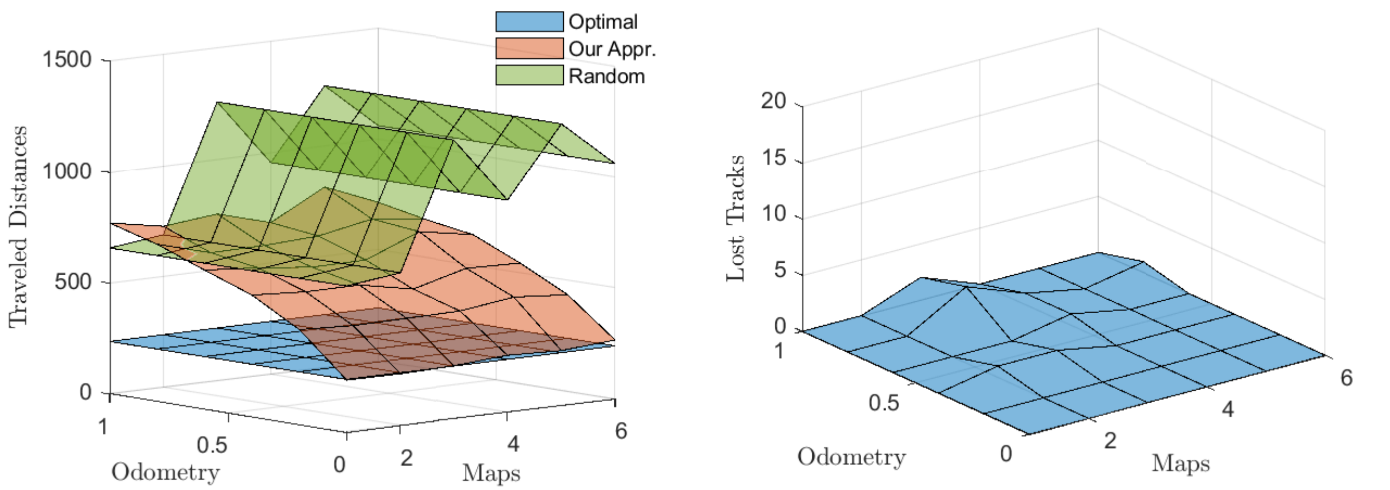
We evaluated our approach in different challenging simulation scenarios based on an odometry and a velocity motion model from [19] where the model parameters have been determined using Maximum Likelihood Estimation [20] on data recorded with a real robot, a Viking MI 422P (Figure 2). This enables realistic simulation studies for analyzing the performance of the proposed method in detail. The simulated robot is only equipped with two in-/outside area detectors for relocalization. A detailed description of the robots setup parameters is given in the appendix Section V-B.

A. Evaluation Criteria

As measurement of the performance, we use the traveled distance required to reach a certain total coverage percentage which in our case is a coverage value of $95\,\%$. In total, the proposed method was evaluated on 6 different maps with different degrees of complexity but same size of coverage space $A = 50\,m^2$, see Figure 11, and for different levels of odometry errors. Here, an odometry level error of one signals a full odometry error as represented by the odometry model parameters from Table II and an odometry level error of zero signals no odometry error at all.
The optimal traveled distance traversing over the centers of the grid map cells to reach $95\,\%$ coverage for a coverage space of $A = 50\,m^2$ can be determined as $T_{\text{opt}} = 237.4\,m$, see appendix Section V-A. Based on this optimal distance, we can define a optimality criterion for our algorithm as

$$\text{Opt} = 1 - \frac{T - T_{\text{opt}}}{T_{\text{rand}} - T_{\text{opt}}}, \tag{16}$$

where T is the average traveled distance our method required and T_{rand} the average traveled distance a random walk pattern required. Here, a value close to one signals a performance



(a) Comparison between optimal, random walk and the here proposed path planning algorithms.

(b) Trials where the particle filter has lost track of the robots pose estimate during path execution.

Fig. 8: Comparison of the here proposed complete coverage method with the commonly used random walk approach and the optimal approach. For the optimal approach the exact pose of the robot is assumed to be known. For the evaluation, the algorithms have been tested in different maps shown in Figure 11, and with different odometry error levels.

TABLE I: Algorithm performance based on Equation (16).

Maps	Odometry Error 0.0	0.2	0.4	0.6	0.8	1.0
1	1.00	0.54	0.26	0.07	-0.12	-0.26
2	1.00	0.60	0.34	0.17	0.10	-0.08
3	0.99	0.82	0.70	0.66	0.60	0.49
4	0.97	0.74	0.65	0.51	0.41	0.38
5	0.99	0.79	0.69	0.57	0.54	0.44
6	0.96	0.76	0.62	0.51	0.47	0.42

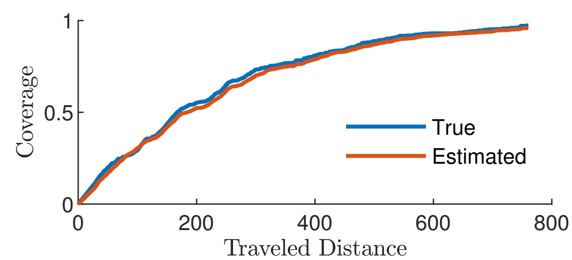


Fig. 9: Comparison between the true and the estimated coverage with respect to the traveled distance of the robot for an odometry level of 1.0 and map 4.

close to the optimal pattern where a value close to zero signals a performance close to a random pattern. A negative value therefore shows a worse performance than a random walk pattern.

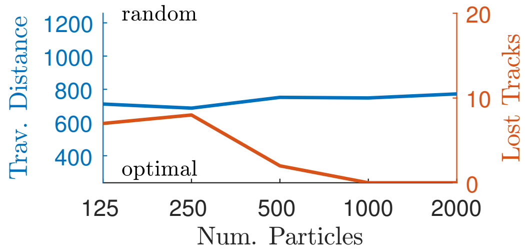
B. *Statistical Evaluation*

For our simulation study, we had 20 runs performed for each pair of map-odometry error level. In total, 6×6 different combination pairs were evaluated with respect to their traveled distance after reaching the $95\,\%$ total coverage level. Since our approach is probabilistic, we averaged the 20 runs recorded for each map-odometry error level pairing. The method parameters used for the evaluation are tabulated in the appendix Section V-B.

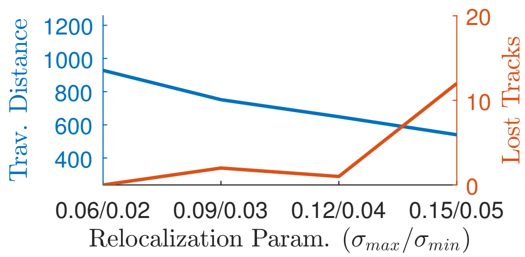
In Figure 8a, the calculated average traveled distances are shown together with the optimal traveled distances and the average traveled distances reached with a random walk pattern. Additionally, in Table I the performance of our method according to Equation (16) is presented. The evaluation shows, that our method in general outperforms the commonly used random walk pattern except for the most simplistic maps under high odometry error. A strong correlation between the performance of our method and the odometry error level can be found such that with lower odometry error our approach performs better and approaches the optimal performance. This is to be expected, since less odometry error allows for better path performance and requires less relocalization. In addition, the complexity of the map favors our approach, as the random walk pattern has trouble covering narrow areas of the map.

C. *Coverage Tracking*

The robot should be able to estimate the current coverage of the workspace as efficiently as possible. Thus, the correlation between the true coverage and the estimated coverage by our approach is of importance. In Figure 9, a comparison between both values is shown for map 4 and an odometry error level of 1.0. The estimated coverage always slightly underestimates the true coverage and thus can be used to conservatively estimate the true coverage.



(a) Effect of the number of particles on the method performance.



(b) Effect of the relocalization on the method performance.

Fig. 10: Algorithm performance (traveled distance, lost tracks) with respect to the number of particles and the relocalization parameter σ_{max}, σ_{min}.

D. Particle Filter Performance Analysis

One disadvantage of our approach is the use of the particle filter, as it sometimes looses track of the pose estimate, Figure 8b, which leads to an abort of the path planning execution. This happens especially with increasing odometry error. Possible error sources, according to [19], are:
(1) The approximation error of the probability distribution due to the finite number of particles used, (2) the approximation error induced by the randomness of the resampling phase, (3) the divergence of the proposal and target distribution if only deterministic measurements are available and (4) the particle deprivation problem, where no particles might be in the vicinity of the correct state.
In general, a larger number of particles used are beneficial for reducing most of the mentioned error sources but comes with a higher computational burden. Nevertheless, an increasing number of particles reduces the number of lost tracks significantly, as shown in Figure 10a. Other measures to be taken are the use of additional sensors to either improve the odometry error, e.g. IMU and odometry sensor fusion, or the external sensory information. Also, a boundary transition zone can be defined to reduce the deterministic character of the in-/outside measurements. Additionally, the parameters for relocalization, σ_{max}, σ_{min}, can be adjusted, such that the robot more often relocalize itself. This reduces the number of lost tracks but also reduces the performance of the method, as shown in Figure 10b.

IV. Conclusion

We proposed a probabilistic approach for complete coverage path planning with low-cost systems. Our method outperforms the commonly used random walk pattern while coping with high pose uncertainties due to only limited sensing capabilities, e.g. only in-/outside area detectors. We analyzed the performance of our method in different challenging scenarios and with different odometry accuracy levels. Thereby, the method proved to be efficient for complete coverage path planning, especially for complex environments. For the future, we will test the robots performance in a real outdoor setting.

V. Appendix

A. Optimal traveled Distance

The optimal traveled distance, assuming the centers of each grid cell should be traversed, can be determined as the number of cells visited multiplied by the length between two grid cell centers, thus the reciprocal of the resolution of the grid. Given a grid map with an area to cover A and a resolution of r, the total number of cells to cover is

$$\text{cells} = Ar^2$$

and the distance required to traverse over all cell centers, thus to reach $100\,\%$ coverage, is then

$$T_{\text{opt}} = \frac{1}{r}\text{cells} = Ar.$$

For this calculation, we assumed that it is possible to find a path along all cell centers without traversing diagonal or revisiting cells. This assumption might not hold always and thus the calculated distance reflects more a lower bound.

B. Robot Setup and Hyper-Parameters used for Simulation

The robot used for simulation is a common autonomous lawn mower, a Viking MI 422P. This robot is a differential drive robot with odometry encoders and two binary in-/outside area detectors at the left and right front. The odometry accuracy has been evaluated performing measurements on a real robot and determining accurate odometry model parameters, Table II. An overview over all parameters for the evaluation of our proposed method is given in Table II.

TABLE II: Robot Setup and Method Parameters

Robot Setup	**Kin. Model**		**Odom. Model**		**Other**	
	α_1	0.011230	a_1	0.002361	max. lin. Vel.	$0.6\,\text{ms}^{-1}$
	α_2	0.003417	a_1	0.000346	max. ang. Vel.	$0.3\,\text{s}^{-1}$
	α_3	0.193604	a_1	0.000223	system frequ.	$20\,\text{Hz}$
	α_4	0.180664	a_1	0.000069		

Method	**Particle Filter**		**Path Planner & Map**			
	N	500	Resolution	$5\,\text{m}^{-1}$	E	100
	N_{thr}	$0.7N$	A	10	γ	0.1
			B	1	σ_{max}	0.09
			D	1	σ_{min}	0.03

References

[1] S Lawrence Bellinger. *Self-propelled random motion lawnmower*. US Patent 3,570,227. Mar. 1971.

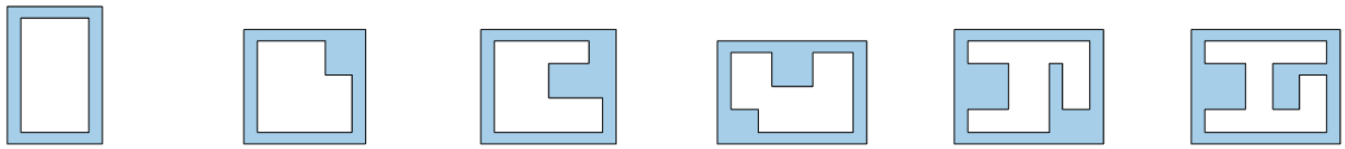


Fig. 11: Different artificial created maps with increasing number of complexity but with the same enclosed area of $A = 50\,m^2$. The maps are numbered from left to right, beginning with Map 1 and ending with Map 6.

[2] N. Rottmann, R. Bruder, A. Schweikard, and E. Rueckert. “Exploiting Chlorophyll Fluorescense for building robust low-cost Mowing Area Detectors”. In: *2020 IEEE SENSORS*. 2020, pp. 1–4. DOI: 10.1109/SENSORS47125.2020.9278858.

[3] N. Rottmann, R. Bruder, A. Schweikard, and E. Rueckert. “A novel Chlorophyll Fluorescence based approach for Mowing Area Classification”. In: *IEEE Sensors Journal* (2020), pp. 1–1. DOI: 10.1109/JSEN.2020.3032722.

[4] Vladimir J Lumelsky, Snehasis Mukhopadhyay, and Kang Sun. “Dynamic path planning in sensor-based terrain acquisition”. In: *IEEE Transactions on Robotics and Automation* 6.4 (1990), pp. 462–472.

[5] Howie Choset and Philippe Pignon. “Coverage path planning: The boustrophedon cellular decomposition”. In: *Field and service robotics*. Springer. 1998, pp. 203–209.

[6] Sylvia C Wong and Bruce A MacDonald. “A topological coverage algorithm for mobile robots”. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*. Vol. 2. IEEE. 2003, pp. 1685–1690.

[7] Hans Moravec and Alberto Elfes. “High resolution maps from wide angle sonar”. In: *Proceedings. 1985 IEEE international conference on robotics and automation*. Vol. 2. IEEE. 1985, pp. 116–121.

[8] Alexander Zelinsky, Ray A Jarvis, JC Byrne, Shinichi Yuta, et al. “Planning paths of complete coverage of an unstructured environment by a mobile robot”. In: *Proceedings of international conference on advanced robotics*. Vol. 13. 1993, pp. 533–538.

[9] Enric Galceran and Marc Carreras. “A survey on coverage path planning for robotics”. In: *Robotics and Autonomous systems* 61.12 (2013), pp. 1258–1276.

[10] Michel Taix, Philippe Souères, Helene Frayssinet, and Lionel Cordesses. “Path planning for complete coverage with agricultural machines”. In: *Field and service robotics*. Springer. 2003, pp. 549–558.

[11] Iwan Ulrich, Francesco Mondada, and J-D Nicoud. “Autonomous vacuum cleaner”. In: *Robotics and autonomous systems* 19.3-4 (1997), pp. 233–245.

[12] Jürgen Hess, Maximilian Beinhofer, Daniel Kuhner, Philipp Ruchti, and Wolfram Burgard. “Poisson-driven dirt maps for efficient robot cleaning”. In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 2245–2250.

[13] Jürgen Hess, Maximilian Beinhofer, and Wolfram Burgard. “A probabilistic approach to high-confidence cleaning guarantees for low-cost cleaning robots”. In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 5600–5605.

[14] Ping-Min Hsu and Chun-Liang Lin. “Optimal planner for lawn mowers”. In: *2010 IEEE 9th International Conference on Cyberntic Intelligent Systems*. IEEE. 2010, pp. 1–7.

[15] Zack J Butler, Alfred A Rizzi, and Ralph L Hollis. “Contact sensor-based coverage of rectilinear environments”. In: *Proceedings of the 1999 IEEE International Symposium on Intelligent Control Intelligent Systems and Semiotics (Cat. No. 99CH37014)*. IEEE. 1999, pp. 266–271.

[16] Simon X Yang and Chaomin Luo. “A neural network approach to complete coverage path planning”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34.1 (2004), pp. 718–724.

[17] Chaomin Luo and Simon X Yang. “A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments”. In: *IEEE Transactions on Neural Networks* 19.7 (2008), pp. 1279–1298.

[18] N. Rottmann, R. Bruder, A. Schweikard, and E. Rueckert. “Loop Closure Detection in Closed Environments”. In: *2019 European Conference on Mobile Robots (ECMR)*. 2019, pp. 1–8. DOI: 10.1109/ECMR.2019.8870938.

[19] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN: 0262201623.

[20] Richard J Rossi. *Mathematical statistics: an introduction to likelihood based inference*. John Wiley & Sons, 2018.

[21] Frank J Massey Jr. “The Kolmogorov-Smirnov test for goodness of fit”. In: *Journal of the American statistical Association* 46.253 (1951), pp. 68–78.

[22] Stephen Grossberg. “Nonlinear neural networks: Principles, mechanisms, and architectures”. In: *Neural networks* 1.1 (1988), pp. 17–61.

[23] Alan L Hodgkin and Andrew F Huxley. “A quantitative description of membrane current and its application to conduction and excitation in nerve”. In: *The Journal of physiology* 117.4 (1952), pp. 500–544.